

**IMAGE PROCESSOR**

Patent Number: JP2000312294  
Publication date: 2000-11-07  
Inventor(s): NAKAJIMA YOSHIYUKI  
Applicant(s): OKI DATA CORP  
Requested Patent: ☐ JP2000312294  
Application Number: JP19990300448 19991022  
Priority Number(s):  
IPC Classification: H04N1/387; G06T3/40; H03M7/30; H04N7/30  
EC Classification:  
Equivalents:

---

**Abstract**

---

**PROBLEM TO BE SOLVED:** To obtain a sharper image by increasing a low resolution of an image.  
**SOLUTION:** A fractal parameter detection section 4 calculates similarity of image data of all partial image consisting of  $8 \times 8$  pixels to detect a fractal parameter. A fractal image generating section 6 generates image data of a fractal image consisting of  $16 \times 16$  pixels based on this fractal parameter. A discrete cosine transform DCT section 9 applies DCT to image data consisting of  $8 \times 8$  pixels segmented from an original image and image data consisting of  $16 \times 16$  pixels segmented from the fractal image. A transform coefficient matrix by applying the DCT to the original image includes much information of low frequency components and a transform coefficient matrix by applying the DCT to the fractal image includes much information of high frequency components. A coefficient data combining section 12 combines the two transform coefficient matrices and an inverse DCT section 13 applies inverse DCT to the combined transform coefficient matrix.

---

Data supplied from the esp@cenet database - 12

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テームト* (参考)
H 0 4 N 1/387	1 0 1	H 0 4 N 1/387	1 0 1 5 B 0 5 7
G 0 6 T 3/40		H 0 3 M 7/30	A 5 C 0 5 9
H 0 3 M 7/30		G 0 6 F 15/66	3 5 5 A 5 C 0 7 6
H 0 4 N 7/30		H 0 4 N 7/133	Z 5 J 0 6 4

審査請求 未請求 請求項の数 5 O L (全 11 頁)

(21) 出願番号 特願平11-300448

(22) 出願日 平成11年10月22日 (1999. 10. 22)

(31) 優先権主張番号 特願平11-45799

(32) 優先日 平成11年2月24日 (1999. 2. 24)

(33) 優先権主張国 日本 (J P)

(71) 出願人 591044164

株式会社沖データ

東京都港区芝浦四丁目11番地22号

(72) 発明者 中島 義幸

東京都港区芝浦四丁目11番地22号 株式会  
社沖データ内

(74) 代理人 100082050

弁理士 佐藤 幸男 (外1名)

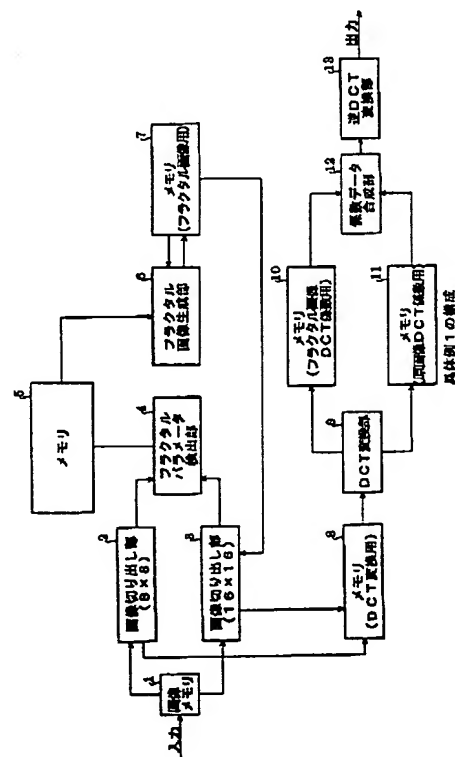
最終頁に続く

(54) 【発明の名称】 画像処理装置

(57) 【要約】

【課題】 低い解像度の画像を高解像度化し、より鮮明な画像を得る。

【解決手段】 フラクタルパラメータ検出部4は、全ての8×8画素の部分画像の画像データに対し、類似度を計算し、フラクタルパラメータを検出する。フラクタル画像生成部6はこのフラクタルパラメータに基づいて16×16画素のフラクタル画像の画像データを生成する。DCT変換部9は、原画像から切り出した8×8画素の画像データ及びフラクタル画像から切り出した16×16の画像データに対してDCTを施す。この原画像のDCTによる変換係数行列には、低周波成分の情報が多く含まれ、フラクタル画像のDCTによる変換係数行列には、高周波成分の情報が多く含まれている。係数データ合成部12はこの2つの変換係数行列を合成し、逆DCT変換部13は合成された変換係数行列に対して逆DCTを行う。



# 【特許請求の範囲】

【請求項1】 直交変換を利用して入力した画像の高解像度化を図る画像処理装置において、  
入力した画像を所定画素数の部分画像に分割して部分画像を切り出す原画像切り出し手段と、

分割された各部分画像よりも大きな画素数を有する領域を入力した画像の中から順次切り出して、切り出した領域のデータに対して所定の変換を行いつつ前記部分画像のデータと比較し、切り出された部分画像のデータに対して自己相似性を有するデータを検出し、部分画像毎に当該データの検出位置及び所定の変換を行ったときの変換パラメータをフラクタルパラメータとして検出するフラクタルパラメータ検出手段と、

検出されたフラクタルパラメータに基づいて部分画像毎に所定のフラクタル処理を行い、部分画像毎に各部分画像よりも大きな画素数を有するフラクタル画像を生成するフラクタル画像生成手段と、

前記各部分画像のデータに対して直交変換を行い、原画像の変換係数行列を生成する第1の直交変換手段と、  
前記フラクタル画像生成手段によって生成されたフラクタル画像のデータに対して直交変換を行い、フラクタル画像の変換係数行列を生成する第2の直交変換手段と、  
該第2の直交変換手段によって生成されたフラクタル画像の変換係数行列のうちの低周波成分の領域に、第1の直交変換手段によって生成された原画像の変換係数行列を上書きして部分画像毎に両変換係数行列を合成する変換係数合成手段と、

該変換係数合成手段によって合成された変換係数行列に対して逆直交変換を行う逆変換手段と、を備えたことを特徴とする画像処理装置。

【請求項2】 前記フラクタルパラメータ検出手段は、入力した画像の中から、部分画像のデータに対して自己相似性を有するデータとして最も類似したデータを検出し、該データの検出位置及び変換パラメータをフラクタルパラメータとして検出するように構成されたことを特徴とする請求項1に記載の画像処理装置。

【請求項3】 前記フラクタルパラメータ検出手段は、入力した画像の中から、部分画像のデータに対して自己相似性を有するデータとして所定の類似度が得られたときのデータを検出し、該データの検出位置及び変換パラメータをフラクタルパラメータとして検出するように構成されたことを特徴とする請求項1に記載の画像処理装置。

【請求項4】 前記変換係数合成手段は、両変換係数行列の各要素をそれぞれ所定の比率に設定して原画像の変換係数行列をフラクタル画像の変換係数行列のうちの低周波成分の領域に上書きするように構成されたことを特徴とする請求項1～請求項3のいずれか1つに記載の画像処理装置。

【請求項5】 フラクタル画像を記憶するフラクタル画

像記憶手段と、入力画像を所定画像サイズに拡大し、初期値として該フラクタル画像記憶手段に記録する初期入力画像記録手段と、を備えたことを特徴とする請求項1～請求項4のいずれか1つに記載の画像処理装置。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、低い解像度の画像を高い解像度の画像に変換する画像処理装置に関する。

【0002】

【従来の技術】例えば、インターネットやデジタルカメラから取得した低い解像度の画像を、高い解像度で出力可能なプリンタやディスプレイ等の出力系に出力する場合、入力画像を高解像度化する技術が必要となってくる。

【0003】このような高解像度化を行う画像処理装置として、従来より、入力画像に対して直交変換の一つである離散コサイン変換 (Discrete Cosine Transform: DCT) を利用したものが知られている。

【0004】図2はかかる従来の画像処理装置の動作を説明するための説明図である。この図2(1)に示すように、従来の画像処理装置では、まず、入力した画像に対し、処理の単位として例えば $8 \times 8$ 画素程度の部分画像に分割し、この部分画像に対してDCTを行い、図2(2)に示すように2次元的な配列に配置された64 ( $= 8 \times 8$ ) 個の変換係数行列の各要素を求める。DCTによって変換された変換係数行列の各要素は各周波数成分の強度を表し、このうち、変換係数行列の左上が低周波成分を表し、右下に行くに従って高周波成分を表すことになる。

【0005】次に、図2(3)に示すように、部分画像よりも大きな画素数の $16 \times 16$ の2次元配列を用意し、この配列の左上の $8 \times 8$ の領域に、入力画像から求めた変換係数行列の各要素を低周波成分として書き込み、高周波成分を示すその他の領域に0を埋める。

【0006】次に、図2(4)に示すように、この2次元的に配置された $16 \times 16$ の変換係数行列に対し、逆DCTを行う。このようにして $16 \times 16$ 画素の画像データが生成され、この画像データを $8 \times 8$ 画素の画像データと置き換える。この操作を画像全体に対して繰り返すことによって画像全体が縦横2倍に変換され、画像の密度が高くなり、結果として解像度の高い画像が得られることになる。

【0007】

【発明が解決しようとする課題】ところで、一般に画像データは、低周波成分の情報を多く含み、DCTによる変換係数行列では、入力した画像データの解像度が低くなるほど、輪郭などの高周波成分が少なくなり、0になる確率が高くなる。

【0008】従来の画像処理装置のように、配列の高周波成分の領域に0を埋めただけでは、たとえ画像の密度

を高くして高解像度化を図ったとしても高周波成分が欠落したままとなり、画像にいわゆるボケが生じ、エッジ部の不鮮明な画像が生成されてしまうという不都合が生じる。従って、より鮮明な画像が得られるような画像処理装置が期待される。

#### 【0009】

【課題を解決するための手段】本発明は以上の点を解決するため次の構成を採用する。

〈構成1〉請求項1の発明に係る画像処理装置は、直交変換を利用して入力した画像の高解像度化を図る画像処理装置において、入力した画像を所定画素数の部分画像に分割して部分画像を切り出す原画像切り出し手段と、分割された各部分画像よりも大きな画素数を有する領域を入力した画像の中から順次切り出して、切り出した領域のデータに対して所定の変換を行いつつ前記部分画像のデータと比較し、切り出された部分画像のデータに対して自己相似性を有するデータを検出し、部分画像毎に当該データの検出位置及び所定の変換を行ったときの交換パラメータをフラクタルパラメータとして検出するフラクタルパラメータ検出手段と、検出されたフラクタルパラメータに基づいて部分画像毎に所定のフラクタル処理を行い、部分画像毎に各部分画像よりも大きな画素数を有するフラクタル画像を生成するフラクタル画像生成手段と、前記各部分画像のデータに対して直交変換を行い、原画像の変換係数行列を生成する第1の直交変換手段と、前記フラクタル画像生成手段によって生成されたフラクタル画像のデータに対して直交変換を行い、フラクタル画像の変換係数行列を生成する第2の直交変換手段と、該第2の直交変換手段によって生成されたフラクタル画像の変換係数行列のうちの低周波成分の領域に、第1の直交変換手段によって生成された原画像の変換係数行列を上書きして部分画像毎に両変換係数行列を合成する変換係数合成手段と、該変換係数合成手段によって合成された変換係数行列に対して逆直交変換を行う逆変換手段と、を備えている。

【0010】〈構成2〉請求項2の発明に係る画像処理装置では、前記フラクタルパラメータ検出手段が、入力した画像の中から、部分画像のデータに対して自己相似性を有するデータとして最も類似したデータを検出し、該データの検出位置及び変換パラメータをフラクタルパラメータとして検出するように構成されている。

【0011】〈構成3〉請求項3の発明に係る画像処理装置では、前記フラクタルパラメータ検出手段が、入力した画像の中から、部分画像のデータに対して自己相似性を有するデータとして所定の類似度が得られたときのデータを検出し、該データの検出位置及び変換パラメータをフラクタルパラメータとして検出するように構成されている。

【0012】〈構成4〉請求項4の発明に係る画像処理装置では、前記変換係数合成手段が、両変換係数行列の

各要素をそれぞれ所定の比率に設定して原画像の変換係数行列をフラクタル画像の変換係数行列のうちの低周波成分の領域に上書きするように構成されている。

【0013】〈構成5〉請求項5の発明に係る画像処理装置では、フラクタル画像を記憶するフラクタル画像記憶手段と、入力画像を所定画像サイズに拡大し、初期値として該フラクタル画像記憶手段に記録する初期入力画像記録手段と、を備えている。

#### 【0014】

【発明の実施の形態】以下、本発明の実施の形態を具体例を用いて説明する。

〈具体例1〉具体例1は、DCTによる高解像度化に、フラクタル画像圧縮法によって生成されたフラクタル画像を利用するようにしたものである。

【0015】ここでフラクタル画像圧縮法について説明する。画像には、画像の一部分の濃度変化に着目したときに、この濃度変化に非常に似通った濃度変化が存在するという性質がある。これを自己相似性といい、フラクタル画像圧縮法は、画像圧縮を、この自己相似性を探索する問題に置き換えて符号化（フラクタルブロック符号化）を行う圧縮法である（画像電子学会誌第26巻第4号（1997年）「フラクタル画像圧縮法の復元画質向上について」参照）。

【0016】このフラクタル画像圧縮法では、着目した領域とは別の領域の画像データに対して縮小変換、回転変換等を行い、得られたデータとの自己相似性を探索し、自己相似性を有するデータの位置及びこのような変換を行ったときの交換パラメータを圧縮データとして符号化する。

【0017】図3はかかるフラクタル画像圧縮法により生成されたフラクタル画像に対して、DCTを施して生成された変換係数行列を示す説明図である。フラクタル画像圧縮法は、高い符号化効率の取得が期待できるため、圧縮技術として利用されているが、図3に示すようにフラクタル画像圧縮法により生成されたフラクタル画像に対してDCTを施すと、生成された変換係数行列には、高周波成分の情報が多く含まれる。

【0018】このフラクタル画像のDCTによる変換係数行列と、低周波成分の情報が多く含まれる原画像のDCTによる変換係数行列とをうまく組み合わせることにより入力画像の高解像度化が期待される。尚、具体例1では、直交変換としてDCTを用いて説明するが、他の直交変換法を用いて周波数分布を得るようにしてもよい。

【0019】次に、画像の高解像度化を行う画像処理装置の具体例1として、その構成を説明する。図1は、具体例1の構成を示すブロック図である。画像メモリ1は、例えば、512×512画素、あるいは1024×1024画素の入力画像を記録する画像メモリである。

【0020】画像切り出し部2は、画像メモリ1上の画

像データから $8 \times 8$ 画素の画像データを切り出す原画像切り出し手段であり、画像切り出し部3は、画像メモリ1上の画像データ又はメモリ7に記録されたフラクタル画像から、より大きなサイズの $16 \times 16$ 画素の画像データを切り出すブロックである。尚、具体例1では、入力画像の画素数を $512 \times 512$ 又は $1024 \times 1024$ 、画像切り出し部2が切り出す部分画像のサイズを $8 \times 8$ 画素、画像切り出し部3が切り出す部分画像のサイズを $16 \times 16$ 画素として説明するが、このサイズに限られるものではない。

【0021】フラクタルパラメータ検出部4は、画像切り出し部3によって切り出された $16 \times 16$ 画素の画像データに対し、所定の変換を行って、画像切り出し部2によって切り出された $8 \times 8$ 画素の部分画像との類似度を自己相似性として計算し、類似した画像データの位置を示すアドレス及び所定の変換に用いた変換パラメータをフラクタルパラメータとして検出し、計算した類似度をフラクタルパラメータと共にメモリ5に出力するブロックである。この画像切り出し部3とフラクタルパラメータ検出部4とがフラクタルパラメータ検出手段に相当する。

【0022】メモリ5は、フラクタルパラメータ検出部4によって求められた各種のパラメータを記録しておくメモリである。尚、記録された各種パラメータは、自己相似性が高くなるに従って更新される。

【0023】メモリ7は、フラクタル画像のデータを記憶するメモリである。メモリ7には、入力画像の縦横2倍の画像データを記憶できるサイズのものをを用いる。

【0024】フラクタル画像生成部6は、記録されたアドレスとフラクタルパラメータとをメモリ5から読み取り、フラクタル画像のデータをメモリ7から読み取って、メモリ5から読み取ったデータに基づいてフラクタル画像を生成し、メモリ7に記録するフラクタル画像生成手段である。

【0025】メモリ8は、DCTを行う対象の画像データを記録するメモリである。DCT変換部9は、メモリ8から画像データを取り出してDCTを行い、取り出した画像データと同じサイズの2次元配列の変換係数行列を出力するブロックであり、このDCT変換部9が第1の直交変換手段及び第2の直交変換手段に相当する。尚、具体例1では、このDCT変換部9を、原画像及びフラクタル画像のDCT変換用として共用したが、それぞれ個別に備えるようにしてもよい。

【0026】メモリ10は、フラクタル処理によって生成された $16 \times 16$ サイズのDCTによる変換係数行列を記録するメモリである。メモリ11は、原画像に基づいて生成された $8 \times 8$ サイズのDCTによる変換係数行列を記録するメモリである。

【0027】係数データ合成部12は、メモリ11に記録された変換係数行列とメモリ10に記録された変換係

数行列とを合成する変換係数合成手段である。逆DCT変換部13は、係数データ合成部12によって合成されたDCTの変換係数行列に対して逆DCTを行い、出力すべき画像データを生成する逆変換手段である。

【0028】〈動作〉次に具体例1の動作を説明する。図4は具体例1の動作を示すフローチャートである。ステップ(図中、ステップを「S」と記す)1では、まず処理対象の画像データを例えば、インターネットやデジタルカメラから入力し、この画像データを画像メモリ1に記録する。

【0029】ステップ2では、入力したすべての画像データに対し、フラクタルパラメータの検出を行う。図5はこの動作を示すフローチャートであり、このフローチャートに基づいてフラクタルパラメータの検出動作について説明する。ステップ21では、カウント値*i*を初期値0にセットする。

【0030】ステップ22では、入力した画像データから $8 \times 8$ 画素の部分画像の画像データR[i]を切り出す。この切り出しは、画像切り出し部2によって重なり合わないように行われる。

【0031】ステップ23では、カウント値*j*を初期値0にセットする。ステップ24では、入力した画像データから $16 \times 16$ 画素の部分画像の画像データD[j]を切り出す。この切り出しは、画像切り出し部3によって1画素ずつずらしながら行われる。

【0032】ステップ25では、類似度best.similarに初期値として、取りうる最大値を設定し、メモリ5に記録する。ステップ26では、切り出した画像データD[j]に対して縮小変換を行う。ここで、縮小変換は画像を重ならないように $2 \times 2$ 画素の部分ブロックに分割し、各ブロック内でR、G、Bの各プレーンに対して平均値を求める処理であり、この平均値を1画素の輝度値とすることによって縦横 $1/2$ 倍の縮小画像が生成される。

【0033】尚、このステップ以降は、フラクタルパラメータ検出部4によって処理される。ステップ27では、縮小変換された画像データD[j]に対してスケール変換を行う。スケール変換は2画像間の輝度値の最大値と最小値の範囲を一致させる処理である。

【0034】ステップ28では、縮小変換及びスケール変換が行われた画像データD[j]に対してシフト変換を行う。シフト変換は画像の輝度値の平均値をシフトして2画像間の輝度値の平均値を一致させる処理である。

【0035】ステップ29では、カウント値*k*を初期値0にセットする。ステップ30では、縮小変換、スケール変換、シフト変換の3通りの処理が行われた画像データD[j]に対し、反転、回転を組み合わせて8種類のアフィン変換を行う。

【0036】ステップ31では、変換された画像データに対して画像データR[i]との類似度similarを計算

する。類似度similarを計算するには、各プレーン、画像全体に対して2画像間の対応する画素の輝度値の差を求め、この総和を計算する。そして、この総和を類似度similarとする。この類似度best.similarが小さいほど2画像間の自己相似性が高いことを示している。

【0037】尚、前述した文献の式(1)に記載されているように、例えば平均自乗誤差法を用いてこの類似度を計算することもできる。ステップ32では、計算した類似度similarを類似度best.similarと比較する。

【0038】そして、計算した類似度similarが類似度best.similar以上のときは、メモリ5に記録したデータを更新せずにステップ34に進む。また、計算した類似度similarが類似度best.similarよりも小さいときは、ステップ33に進み、メモリ5に記録したアドレスbest.x、best.y、スケール量best.scale、シフト量best.shift、アフィン変換量best.affine、類似度best.similarを、それぞれこのときのアドレスx、y、スケール量scale、シフト量shift、アフィン変換量affine、類似度similarで書き換えてステップ34に進む。

【0039】ステップ34では、カウント値kをインクリメントする。このカウント値kが8未満のときは、ステップ30に戻り、同じようにアフィン変換を行う。カウント値kが8になったとき、即ち、8種類のアフィン変換がすべて行われたときは、ステップ36に進む。

【0040】カウント値jが所定値M未満のときは、カウント値jをインクリメントしてからステップ24に進み、次の画像データD[j]に対し、ステップ24～35の処理を繰り返す。

【0041】カウント値jが所定値Mになったとき、即ち、ある画像データR[i]に対し、すべての画像データD[j]との類似度の計算が終わったときは、ステップ37に進む。

【0042】カウント値iが所定値N(=入力画像を重ならない8×8の部分画像に分割したときの部分画像の数)未満のときは、カウント値iをインクリメントしてからステップ21に戻り、順次、次の画像データR[i]に対してステップ22～36の処理を繰り返す。

【0043】そして、カウント値iが所定値Nになったとき、即ち、すべての画像データR[i]に対するフラクタルパラメータの検出が終了したとき、このルーチンが終了する。

【0044】これにより、すべての画像データR[i]に対し、最も類似している画像データD[j]の座標、スケール量、シフト量、アフィン変換量がメモリ5に記録される。そしてこのデータはパラメータのテーブルfractal\_tblの形式で保存される。

【0045】次に図4のステップ3に進む。ステップ3では、検出されたフラクタルパラメータに基づいてフラクタル処理を行い、縦横2倍のフラクタル画像を生成する。この処理はフラクタル画像生成部6によって行われ

る。

【0046】図6は、このフラクタル画像生成部6の動作を示すフローチャートであり、このフローチャートに基づいてフラクタル画像生成部6の動作を説明する。ステップ41では、まずフラクタル処理を行うための初期画像Fを生成する。前述のように、この初期画像Fの画像サイズは入力画像の縦横2倍であり、初期画像データとして、全画素につき例えば中間値であるR(Red)=128、G(Green)=128、B(Blue)=128を記録する。

【0047】ステップ42では、カウント値iを初期値0にセットする。ステップ43では、初期画像Fから部分画像の画像データを切り出す。この画像データの切り出しは、座標(fractal\_tbl[i].x×2, fractal\_tbl[i].y×2)を起点として、32×32画素のサイズで行う。

【0048】ステップ44～47では、切り出した画像データに対し、テーブルfractal\_tblの各パラメータを参照して縮小変換、スケール変換、シフト変換、アフィン変換を行う。この変換方法は、フラクタルパラメータ検出部4によって行われた変換方法と同じである。

【0049】ステップ48では、この各種変換を行って生成された画像をフラクタル画像としてi番目の16×16サイズの部分に貼り付ける。ステップ49では、カウント値iをインクリメントする。

【0050】カウント値iがN未満のときは、ステップ43～49の処理を繰り返す。カウント値iがNになったとき、即ち、全てのフラクタルパラメータに対する処理が終了したときは、入力画像の縦横2倍サイズのフラクタル画像が改めて生成され、このときは、ステップ51に進む。そして、終了条件となったときは、フラクタル画像生成部6の動作が終了し、図4のステップ4に進む。

【0051】ステップ4では、カウント値iを初期値0にセットする。ステップ5では、入力した画像データから8×8画素の画像データI[i]を切り出す。この切り出しは、画像切り出し部2によって行われる。

【0052】図7は具体例1のデータの遷移を示す説明図であり、図7(1)は、画像切り出し部2によって入力画像から切り出された8×8画素の画像データを示す。ステップ6では、メモリ7に記憶されたフラクタル画像から重なり合わない16×16画素のフラクタル画像データF[i]を切り出す。この切り出しは、画像切り出し部3によって行われる。

【0053】図7(3)は、画像切り出し部3によって切り出されたフラクタル画像データF[i]を示し、この図7(3)に示すフラクタル画像データF[i]は、図7(1)に示す8×8の入力画像データI[i]に対応している。

【0054】ステップ7では、切り出した画像データI

[i] に対してDCTを行い、 $8 \times 8$ サイズの変換係数行列を生成し、メモリ10に記憶する。図7(2)は、生成された $8 \times 8$ サイズの変換係数行列を示す。図2(2)に示すように、この変換係数行列には低周波成分の情報が多く含まれている。

【0055】ステップ8では、切り出した画像データF[i]に対してDCTを行い、 $16 \times 16$ サイズの変換係数行列を生成し、メモリ11に記憶する。図7(4)は、生成された $16 \times 16$ サイズの変換係数行列を示す。図3に示すように、この変換係数行列には高周波成分の情報が多く含まれている。尚、このステップ7、8の処理は、DCT変換部9によって行われる。

【0056】ステップ9では、図7(5)に示すように、メモリ11に記憶され、2次元的に配置された $16 \times 16$ の変換係数行列の左上の第2象限に相当する $8 \times 8$ のサイズの領域、即ち、低周波成分の領域に、メモリ10に記憶された $8 \times 8$ の変換係数行列を上書きする。これによって2つの変換係数行列が合成される。この処理は、係数データ合成部12によって行われる。

【0057】ステップ10では、この合成された変換係数行列に対して逆DCTを行う。この処理は、逆DCT変換部13によって行われる。図7(6)は逆DCTが行われて生成された $16 \times 16$ 画素の画像データを示し、この画像データには低周波成分から高周波成分までの情報が含まれることになる。

【0058】ステップ11では、カウント値iをインクリメントする。ステップ12では、カウント値iを所定値N(=入力画像を重ねのりない $8 \times 8$ の部分画像に分割したときの部分画像の数)と比較する。カウント値iが所定値N未満のときは、次の画像データI[i]に対し、ステップ5～11の処理を繰り返す。カウント値iが所定値Nになったとき、即ち、すべての画像データI[i]に対して処理が終了したときはこのルーチンを終了する。

【0059】そして、元の画像に対してこの画像を置き換えていけば、エッジ部まで鮮明な画像に高解像度化され、高い解像度を有するプリンタやディスプレイ等の出力系に出力される。

【0060】〈具体例1の効果〉以上、説明したように具体例1によれば、全ての $8 \times 8$ 画素の部分画像の画像データに対し、最も類似している $16 \times 16$ 画素のフラクタル画像の画像データを生成し、ともにDCTを施して、生成された両変換係数行列を合成し、逆DCTを行うことにより、出力された画像には、入力された画像には存在していなかったエッジなどの高周波成分が多く含まれるようになり、エッジ部の鮮明な画像を再現することができる。従って、高い解像度のプリンタやディスプレイ等の出力系に、最も適した画像を出力することができる。

【0061】〈具体例2〉具体例2は、類似度が所定の

閾所値以下になったときに類似度の計算を終了させるようにしたものである。尚、具体例2は具体例1と動作が異なるだけで、その構成は、具体例1の構成と同一であるので、同一符号を付してその説明を省略する。

【0062】〈動作〉次に具体例2の動作を説明する。図8は、具体例2のフラクタルパラメータ検出の動作を示すフローチャートである。具体例2では、具体例1のステップ33とステップ34との間でステップ61の処理を行う。尚、その他のステップについては具体例1と同一の動作なので、同一ステップ符号を付して説明を省略する。

【0063】このステップ61では、メモリ5に記録された類似度similarを所定の閾値threadと比較する。そして、この類似度similarが閾値threadよりも小さくなったときは、フラクタルパラメータ検出の処理を終了させる。これにより、一定の類似度が得られれば、フラクタルパラメータ検出の処理が終了するので、フラクタルパラメータ検出の際の探索範囲が減少し、画像生成が高速化する。

【0064】〈具体例2の効果〉以上、説明したように具体例2によれば、一定の類似度が得られればフラクタルパラメータの検出を終了させるので、より高速に画像生成を行うことができる。

【0065】〈具体例3〉具体例3は、係数データ合成部が、生成された入力画像の変換係数行列の各要素とフラクタル画像の変換係数行列の各要素とを所定の比率に設定して加算するようにしたものである。尚、具体例3は具体例1と動作が異なるだけで、その構成は、具体例1の構成と同一であるので、同一符号を付してその説明を省略する。

【0066】〈動作〉次に具体例3の動作を説明する。具体例3では、係数データ合成部12が、メモリ11に記憶されている入力画像の $8 \times 8$ サイズの変換係数行列を、0領域で $16 \times 16$ サイズの変換係数行列に拡張し、生成された入力画像の変換係数行列の各要素とフラクタル画像の変換係数行列の各要素とを次式に従って加算する。

$$D0 = (1 - \alpha) Di + \alpha Df$$

但し、D0：新しく生成される変換係数行列

$\alpha$ ：加算時に設定された比率

Di：DCT変換係数行列

Df：フラクタル画像変換係数行列

【0067】この比率 $\alpha$ は、両変換係数行列の各要素に対して与えられる。図9は、この比率 $\alpha$ の設定例を示す説明図である。この図9の直線で示すように、比率 $\alpha$ は両変換係数行列の周波数に比例して大きくなるように設定される。即ち、周波数が低いときは比率 $\alpha$ は小さな値となり、高周波になるに従って大きくなる。但し、生成された出力画像に応じて、この比率 $\alpha$ を例えば破線のように補正することもできる。

【0068】〈具体例3の効果〉以上、説明したように具体例3によれば、生成された入力画像の変換係数行列の各要素とフラクタル画像の変換係数行列の各要素とを、低周波成分ほど小さく、高周波成分になるに従って大きくなるように比率 $\alpha$ に設定して加算するようにしたので、よりエッジ部の鮮明な画像を再現することが可能となる。

【0069】〈具体例4〉具体例4は、入力画像を所定画像サイズに拡大し、初期値としてフラクタル画像記憶用のメモリに記録するようにしたものである。

【0070】図10は具体例4の構成を示すブロック図である。具体例4では、画像メモリ1の画像を単純に拡大してフラクタル画像用のメモリ7に入力する単純拡大処理部14が備えられている。尚、メモリ7がフラクタル画像記憶手段に相当し、単純拡大処理部14が初期入力画像記録手段に相当する。尚、具体例1と同一要素については、同一符号を付してその説明を省略する。

【0071】〈動作〉次に具体例4の動作を説明する。具体例1では、フラクタル画像記憶用のメモリ7に、初期値として中間値であるグレースケールの画像を入力したが、具体例4では、入力画像が単純拡大処理部14によって所定画像サイズに拡大され、この画像が初期値としてメモリ7に入力される。尚、この拡大率は、フラクタル画像が縦横2倍に拡大されるので、これと同じ率に設定される。

【0072】この拡大された画像を初期値として用いることにより、フラクタルパラメータの検出に用いた情報をそのまま使用できるため、フラクタル画像への収束が早まる。

【0073】〈具体例4の効果〉以上、説明したように具体例4によれば、実際の画像である入力画像を単純拡大して初期値としてメモリ7に記録することにより、相似度の高いフラクタル画像に、より高速に収束させることができる。

【0074】従って、具体例2のように、一定の類似度が得られたときにフラクタルパラメータの検出を終了させるようにした場合でも、より類似度の高いフラクタル画像を得ることができる。

【図面の簡単な説明】

【図1】本発明に係る画像処理装置の具体例1の構成を示すブロック図である。

【図2】従来の画像処理装置の動作を説明するための説明図である。

【図3】具体例1の構造分解されたフラクタル画像について説明するための説明図である。

【図4】具体例1の動作を示すフローチャートである。

【図5】具体例1のフラクタル検出の動作を示すフローチャートである。

【図6】具体例1のフラクタル画像生成部の動作を示すフローチャートである。

【図7】具体例1の動作を説明するための説明図である。

【図8】本発明に係る画像処理装置の具体例2のフラクタル検出の動作を示すフローチャートである。

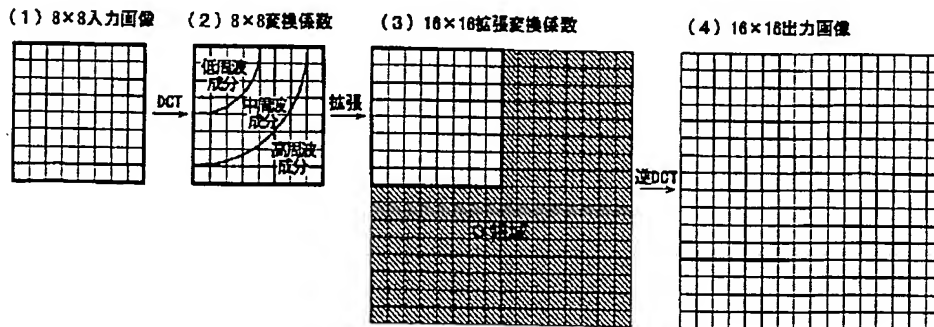
【図9】本発明に係る画像処理装置の具体例3の変換係数行列を合成するときの比率の設定例を示す説明図である。

【図10】本発明に係る画像処理装置の具体例4の構成を示すブロック図である。

【符号の説明】

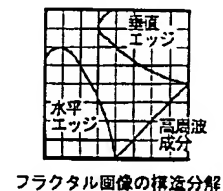
- 2 画像切り出し部(8×8)
- 3 画像切り出し部(16×16)
- 4 フラクタルパラメータ検出部
- 9 DCT変換部
- 12 係数データ合成部
- 13 逆DCT変換部
- 14 単純拡大処理部

【図2】



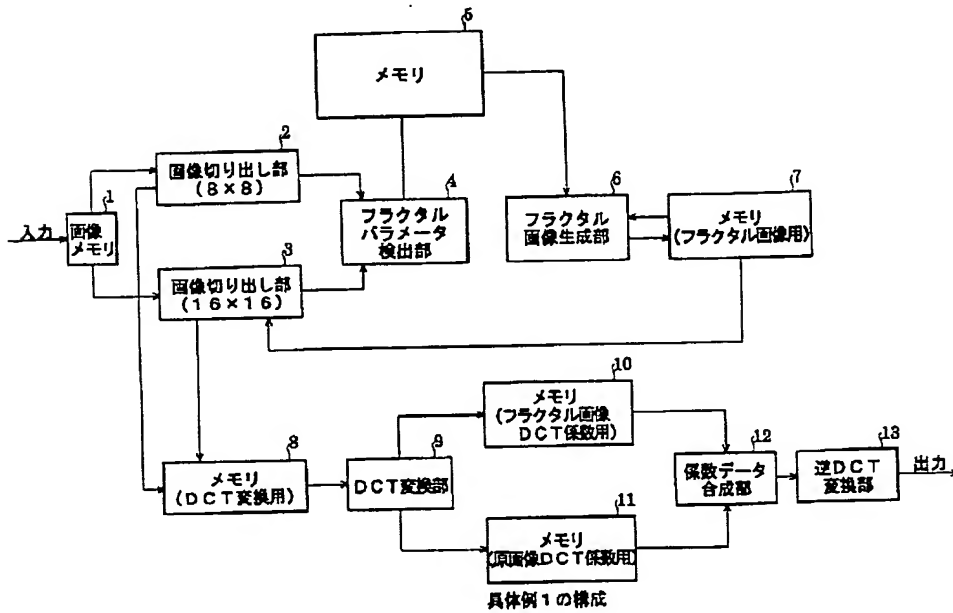
従来のデータの遷移

【図3】



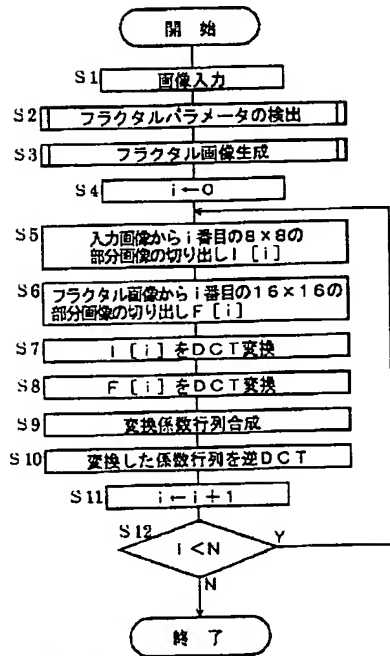
フラクタル画像の構造分解

【図1】



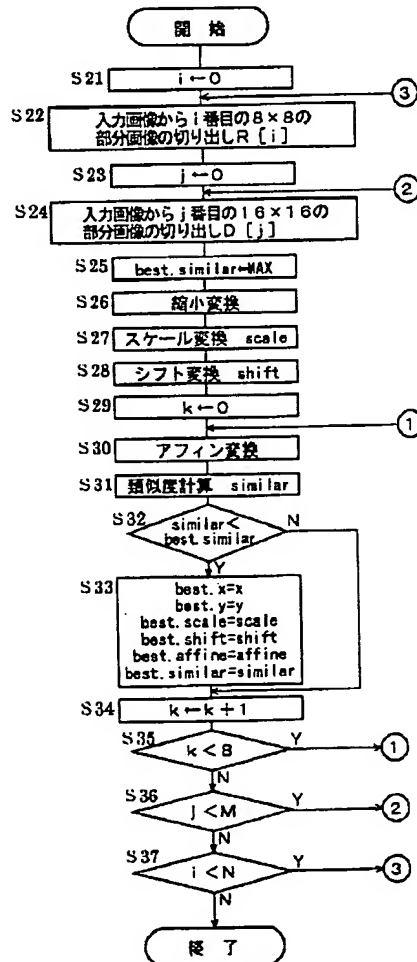
具体例1の構成

【図4】



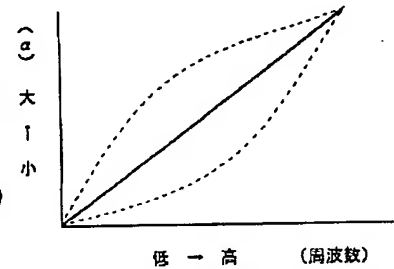
具体例1の動作を示すフローチャート

【図5】



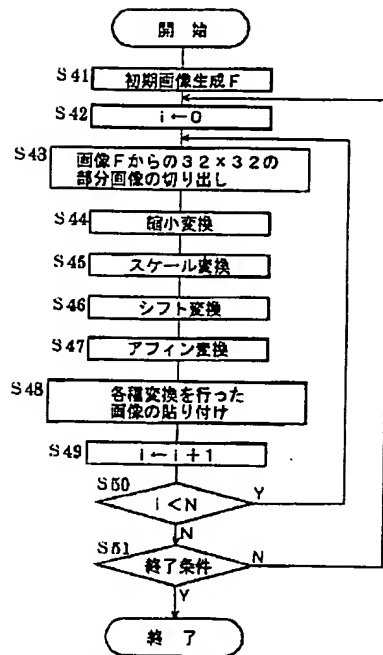
具体例1のフラクタル検出の動作を示すフローチャート

【図9】



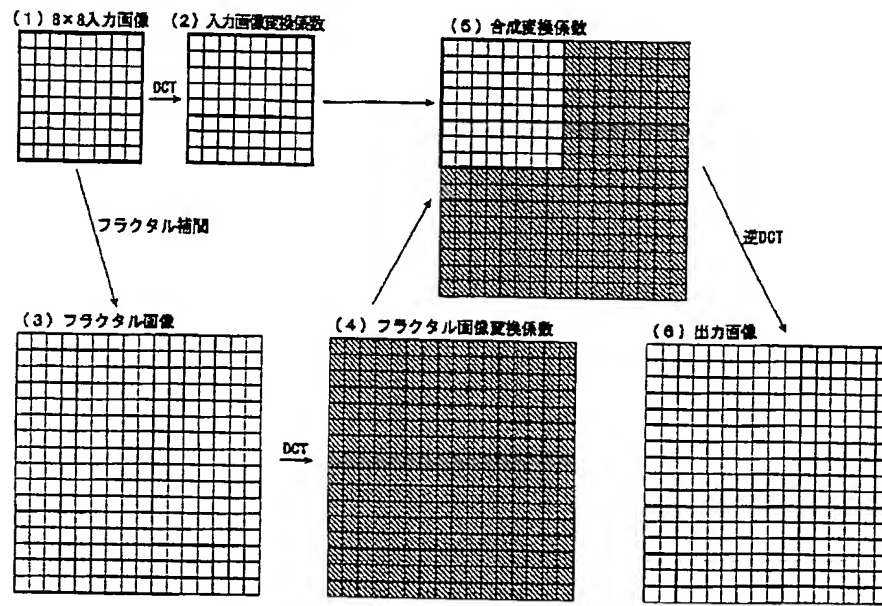
具体例3の動作を示す説明図

【図6】



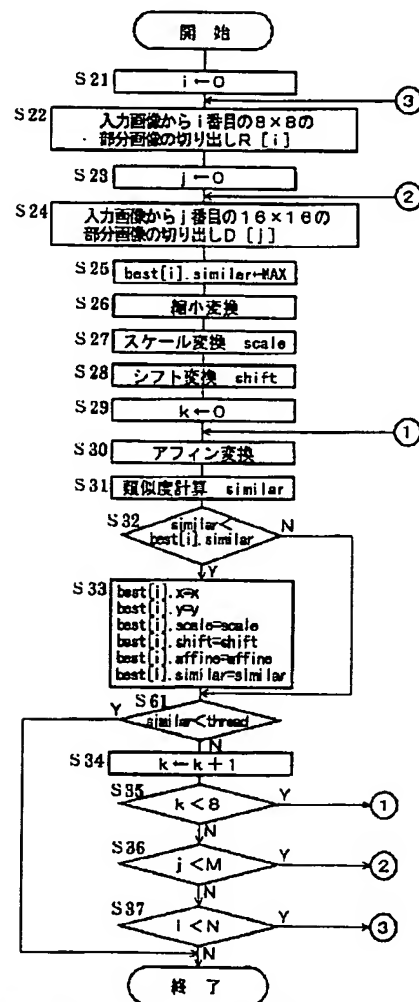
フラクタル画像生成部の動作を示すフローチャート

【図7】



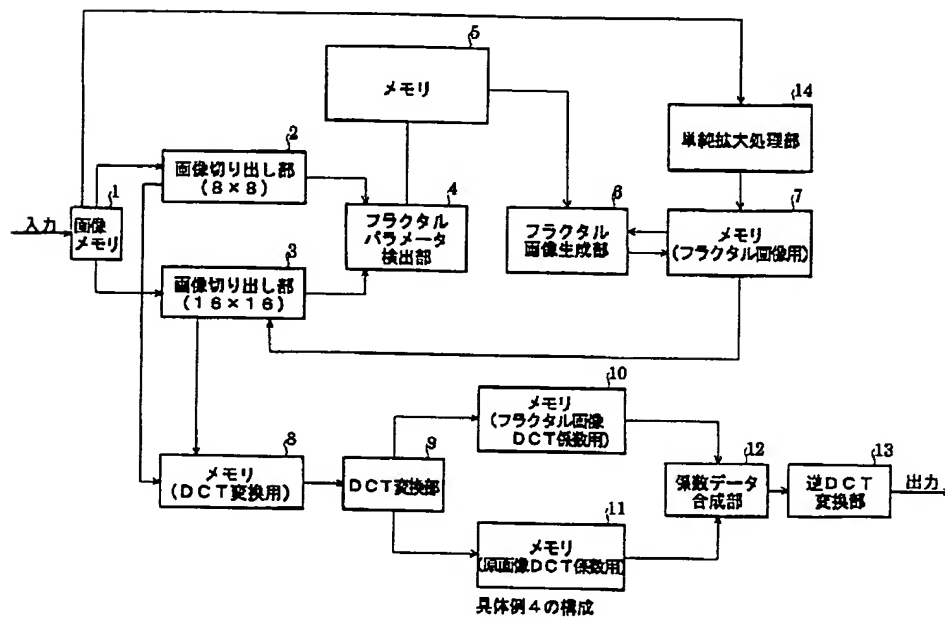
具体例1のデータの遷移

【図8】



具体例2のフラクタル検出の動作を示すフローチャート

【図10】



フロントページの続き

Fターム(参考) 5B057 BA02 CA08 CA12 CA16 CB08  
 CB12 CB16 CC02 CD01 CD05  
 CE03 CE08 CE09 CG07 CH08  
 CH18 DA08 DA17 DB02 DB09  
 DC34 DC36  
 5C059 KK01 LA00 LC08 MA23 MA43  
 MC26 MC30 PP01 PP04 SS08  
 SS14 UA31  
 5C076 AA21 AA22 AA31 AA36 BA06  
 BA09 BB40 CB04  
 5J064 AA01 BA16 BC01 BD03